

REMARKS

Reconsideration and allowance of this subject application are respectfully requested.

Applicants' representative appreciates the time that Examiner Patel took to discuss this case on January 12, 2004. During that interview, the independent claims and several dependent claims were discussed. Applicants' representative identified and explained differences between those claims and the references applied by the Examiner. This Request for Continued Examination and response have been filed to continue prosecution.

In the final Office Action, the Examiner noted a number of informalities regarding dependencies of claims 8 and 24. Those were corrected by the Amendment after final filed and entered by the Examiner.

Claims 1-2, 6, 8-9, 11-18, 22, 24-25, and 27-30 stand rejected under 35 U.S.C. §103 as being unpatentable over DeRoo and Alexander. This rejection is respectfully traversed.

As admitted by the Examiner, DeRoo fails to teach irreversible blocking of subsequent writing to a protected part of memory after data is initially stored there. As admitted by the Examiner, DeRoo's protected area can be written into, which means that there is no irreversible blocking.

The Examiner turns to Alexander in attempt to remedy this particular deficiency. The Examiner's attention is directed to Figure 7 which shows a register A for storing the

address of the first block to be write-protected. Register B is used to store the number of contiguous blocks to be write-protected. Both registers A and B are rendered inaccessible to prevent alteration of either the start block address or the number of contiguous blocks. Subsequent attempts to rewrite bits in registers A and B are prevented by the state of a one-bit register C that may be set or cleared by writing an appropriate bit to that register.

It is not clear whether the security register C irreversibly blocks subsequent writing into a protected area of memory. Indeed, the one-bit register C is described by Alexander as being alterable:

each lockout fuse is modeled by a one bit register (e.g., register C of Fig. 7) that may be set or cleared by writing an appropriate bit to that register.

Column 7, lines 19-22 suggests that a "set" register C in the secure state may be cleared to permit subsequent writing to portions of memory designated by registers A and B. Thus, it appears that those protected memory portions are not irreversibly blocked from subsequent writing. Accordingly, Alexander does not remedy DeRoo's admitted deficiency.

But there is another deficiency in DeRoo and Alexander. Specifically, independent claims 1 and 17 recite that the processor for executing a program routine stored in the memory "is arranged to necessarily execute a security program routine stored in said protected part of said memory upon start up." The additional layer of security provided by executing a security program stored in a protected part of memory

upon start-up provides a high security level in a relatively simple manner. For example, the security program routine may check whether data in accessible memory locations has been modified, and if so, whether those changes were authorized. As a result, even though a malefactor may be able to alter accessible data, that malefactor would not be able to circumvent the security program routine. Because the processor necessarily executes the security program routine, and because that security program routine can not be altered, (being stored in protected memory), unauthorized changes are immediately detected upon start-up.

The simplicity of this two-prong security approach—irreversible blocking of subsequent writing to a protected part of memory coupled with necessary execution of a security routine program stored in the protected part of memory at start-up—is important. Conventional security approaches try to protect the processor from tampering by placing it in an inaccessible location such as in a chip card. But these conventional security measures are not necessary when using the particular invention defined in claims 1 and 17. A conventional processor can be unprotected as long as it is arranged to necessarily execute the security routine from the protected part of memory at start-up. Moreover, the Examiner should understand that a boot routine is not a security routine. A boot routine merely serves to initialize the system, but it does not have the security features recited in claims 1 and 17.

Assuming for arguments sake that the combination of DeRoo and Alexander could be made, that combination fails to disclose irreversible blocking of subsequent data

writing into a protected part of memory coupled with a processor arranged to necessarily execute a security program routine stored in the protected part of memory upon start-up. But even if Alexander described register C as being physically unalterable after being set to effect irreversible blocking of subsequent writing into a protected portion of memory, it would be inconsistent to combine such a teaching with DeRoo. DeRoo's basic premise is to allow data writing into a protected part of memory. To modify DeRoo so that its very premise is destroyed renders DeRoo inoperable for its intended purpose. The Federal Circuit has admonished that a modification of a reference which renders it inoperable for its intended purpose is inappropriate for an obviousness rejection. *In re Fritch* 972 F.2d 1260, 1265-1266, (Fed. Cir. 1992). The combination of DeRoo and Alexander is therefore improper.

Additional dependent claim features are also not disclosed by the combination of DeRoo and Alexander. For example, the Examiner contends that Alexander teaches the features of claims 8 and 24. Particularly, the Examiner contends that Alexander discloses that the "write line is permanently interrupted," reading this language on an address comparison operation. Applicants have reviewed the text in Alexander at column 2, lines 44-48 and find no teaching of a "write line" in Alexander's EEPROM being "permanently interrupted." Indeed, even when register C is set, it is the data paths to registers A and B which are physically interrupted. A write line used to write data into a specific part of memory is not physically interrupted. In other words, registers A and B in Alexander are not part of the protected portion of memory.

Regarding claims 9 and 25, the write line, which is a fusable link in these claims, is not a line used to write to the registers A and B, but rather a write line to write into a protected part of the memory. See, for example, the first step in claims 8 and 24 from which claims 9 and 25 respectively depend.

Regarding dependent claim 16, the Examiner fails to point out where DeRoo discloses that the communication device is a "bluetooth communication device."

Claims 10 and 26 stand rejected under 35 U.S.C. §103 as being unpatentable over DeRoo in view of Alexander and further in view of U.S. Patent 5,546,561 to Kynett et al. This rejection is respectfully traversed.

The Examiner refers to column 6, lines 1-7 which describe a write state machine controlling a write path and verification circuitry of a memory array 22. Although the write state machine 32 may control times when the memory may be programmed or erased, there is no teaching in Kynett that the write state machine 32 protects the nonvolatile memory so that subsequent writing to a protected part is irreversibly blocked.

The Examiner alleges it would have been obvious to combine the teachings of Kynett with both DeRoo and Alexander because Kynett's state machine "will simplify the process by deceasing [sic] the system throughput," referring to column 2, lines 47-57. Applicants have reviewed this text in column 2 and find no support for the Examiner's contention that the Kynett's state machine would simplify protecting memory from being written into by decreasing system throughput.

Claims 3-5, and 19-21 stand rejected under 35 U.S.C. §103 as being unpatentable over DeRoo, Alexander, and further in view of U.S. Patent 6,401,208 to Davis et al. This rejection is respectfully traversed.

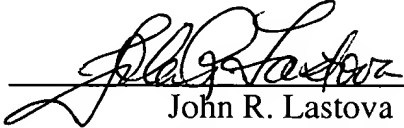
Davis discloses embedding a BIOS certificate and a BIOS signature in the BIOS code in order to authenticate the BIOS code. During an authentication process, the BIOS certificate is decrypted using the root certification key to retrieve a public key of the BIOS signature. Various processings are then performed based on this cryptographic information to determine if there is a match. If the match occurs, the BIOS code has been authenticated. If not, there is no authentication. In contrast, claims 4 and 20 recite "calculating a characteristic parameter for data being checked for changes" in a portion of the memory. In contrast, Davis' system simply determines whether the certification information matches and whether the BIOS code is authentic. No determination is made whether the BIOS code has been changed. The Examiner contends that the characteristic parameter is a checksum, as claimed in claims 5 and 21, and refers generically to Figure 6B in Davis. While there are various cryptographic operations performed, Applicants find no explicit teaching of a checksum being calculated to check for changes in a portion of memory in Figure 6B in Davis.

Applicants respectfully submit that the present application is condition for allowance. An early notice to that effect is earnestly solicited.

MÖLLER et al.
Appl. No. 09/598,173
March 5, 2004

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 
John R. Lastova
Reg. No. 33,149

JRL:at
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100